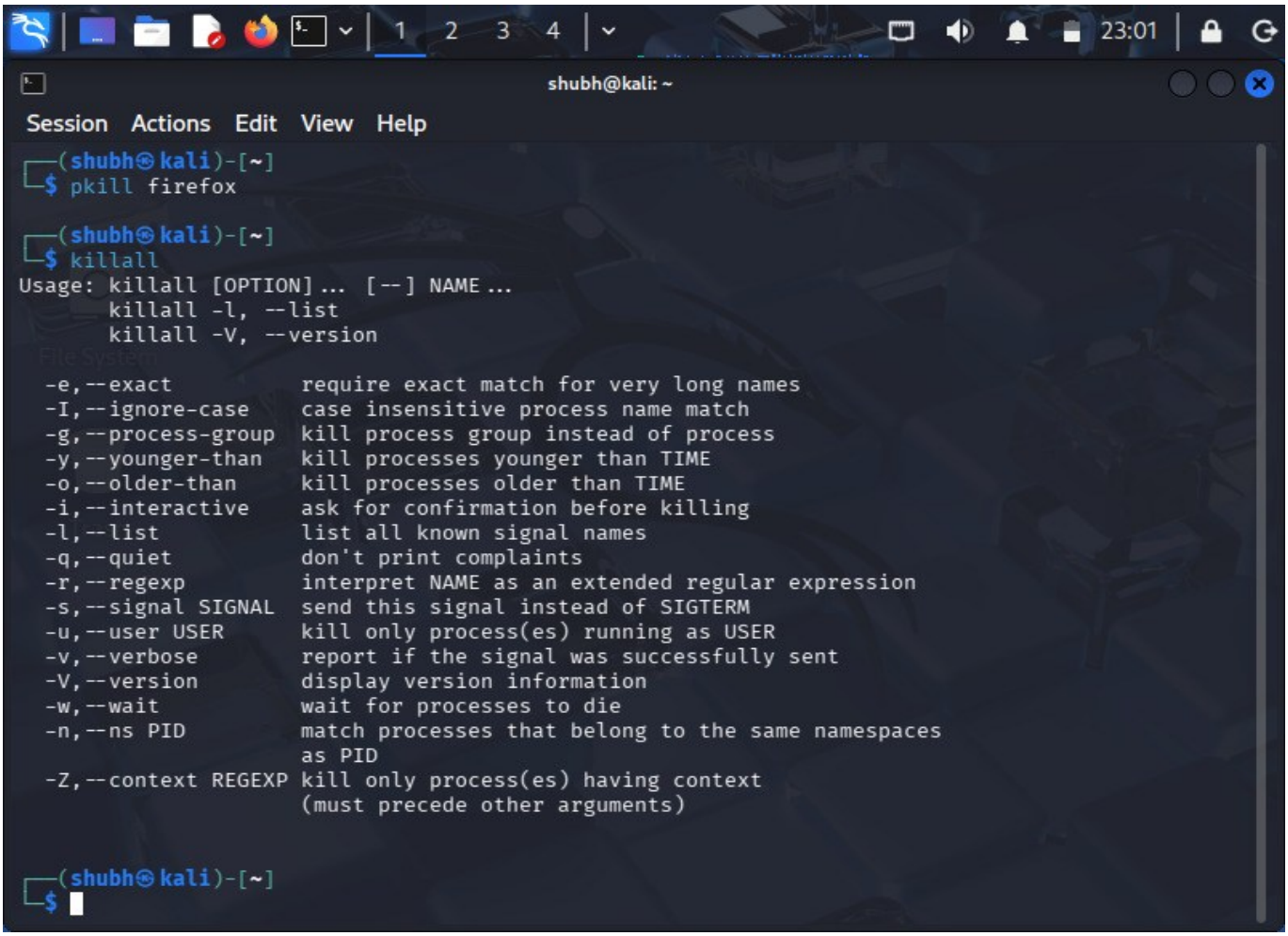
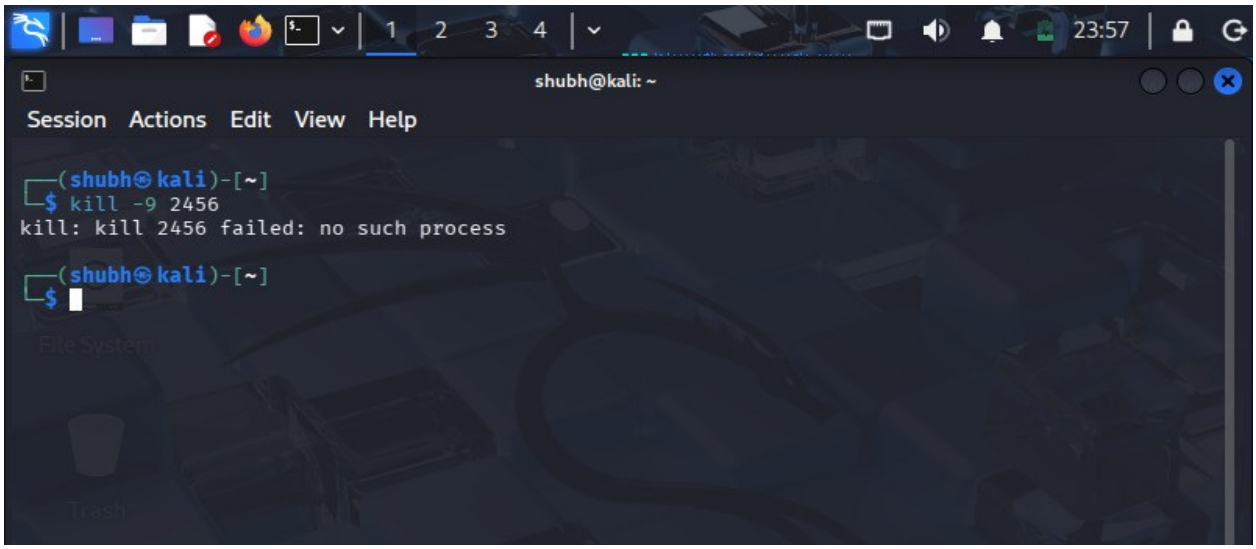
1.Kill Processes by name

2.Kill a process based on the process name .

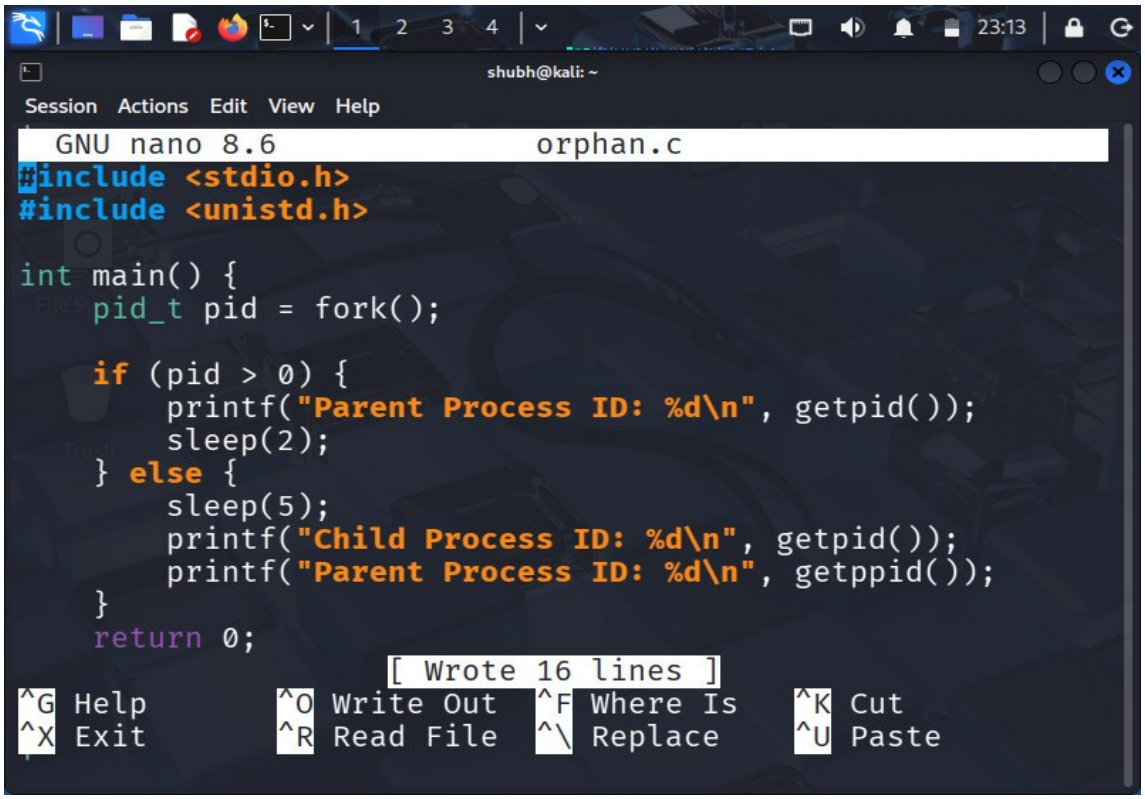
3.Kill a single process at a time with the given process lD

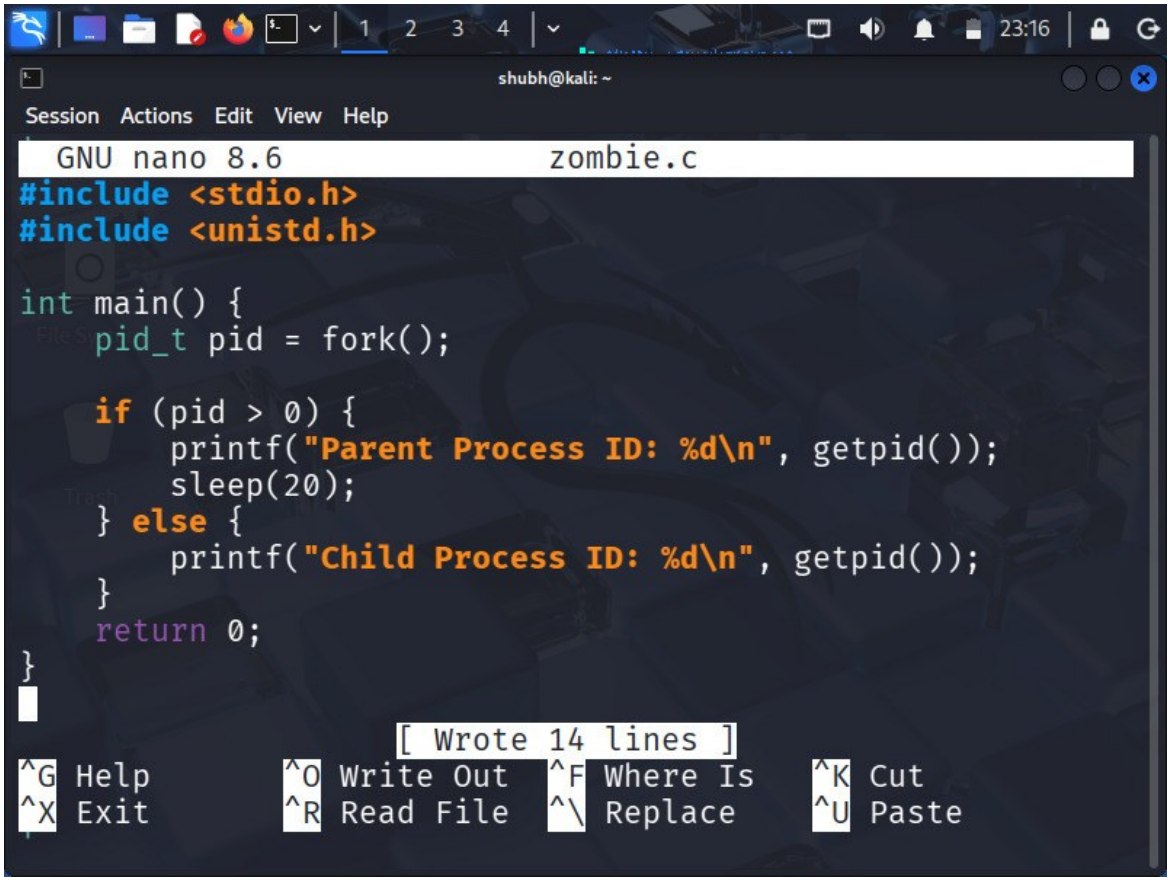




2.Write a program for process creation using C

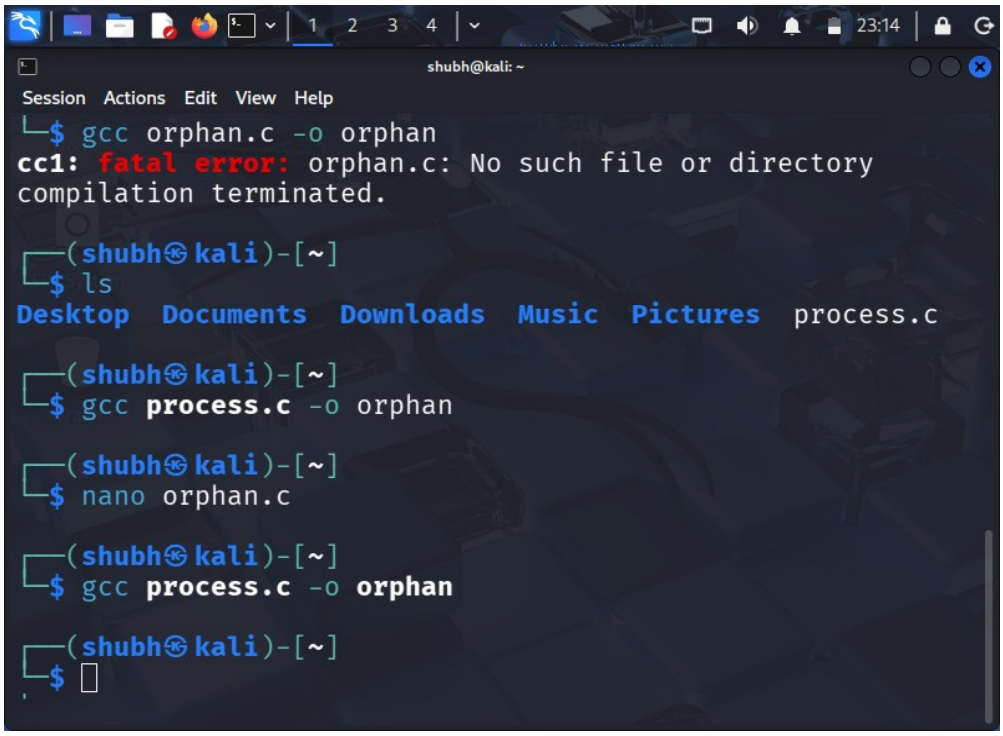
1. Orphan Process 2.Zombie process

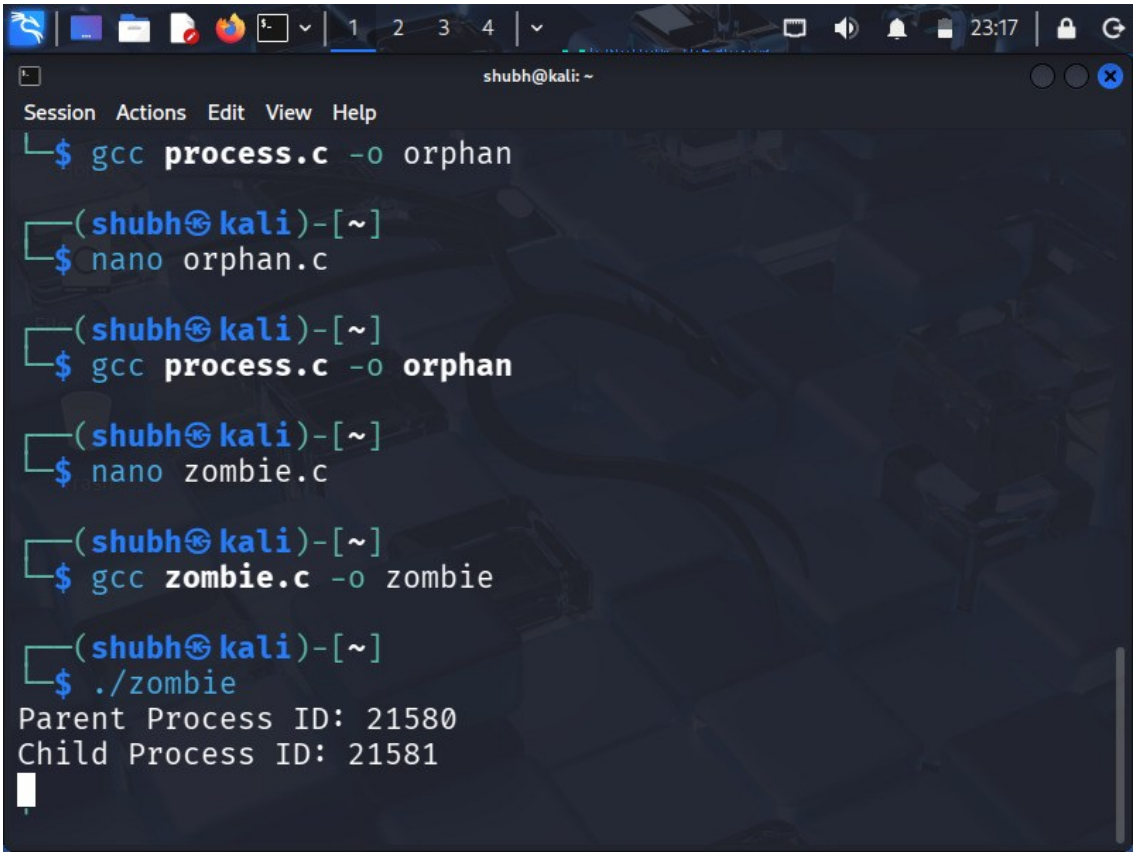


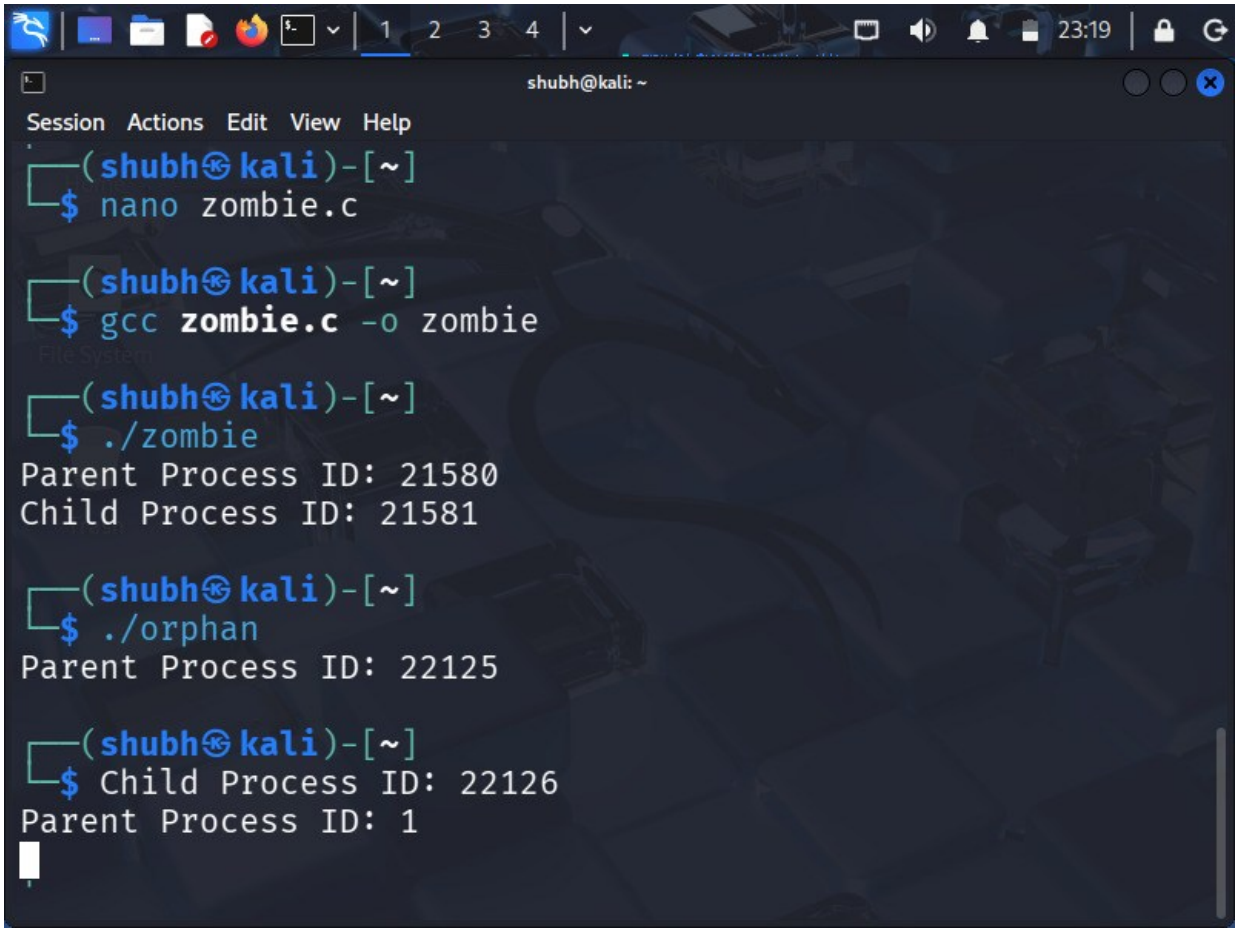


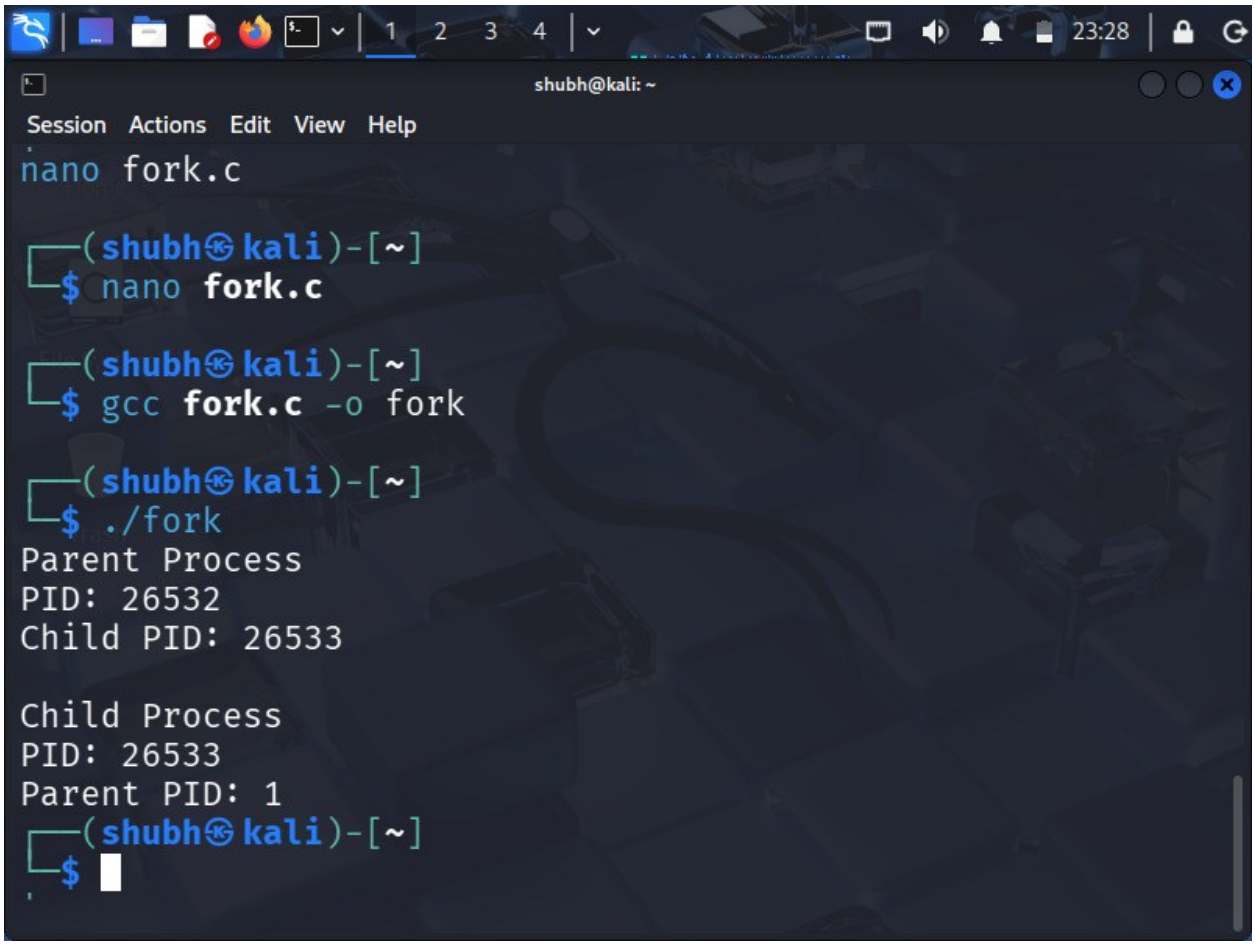
3. Create the process using fork()system cell

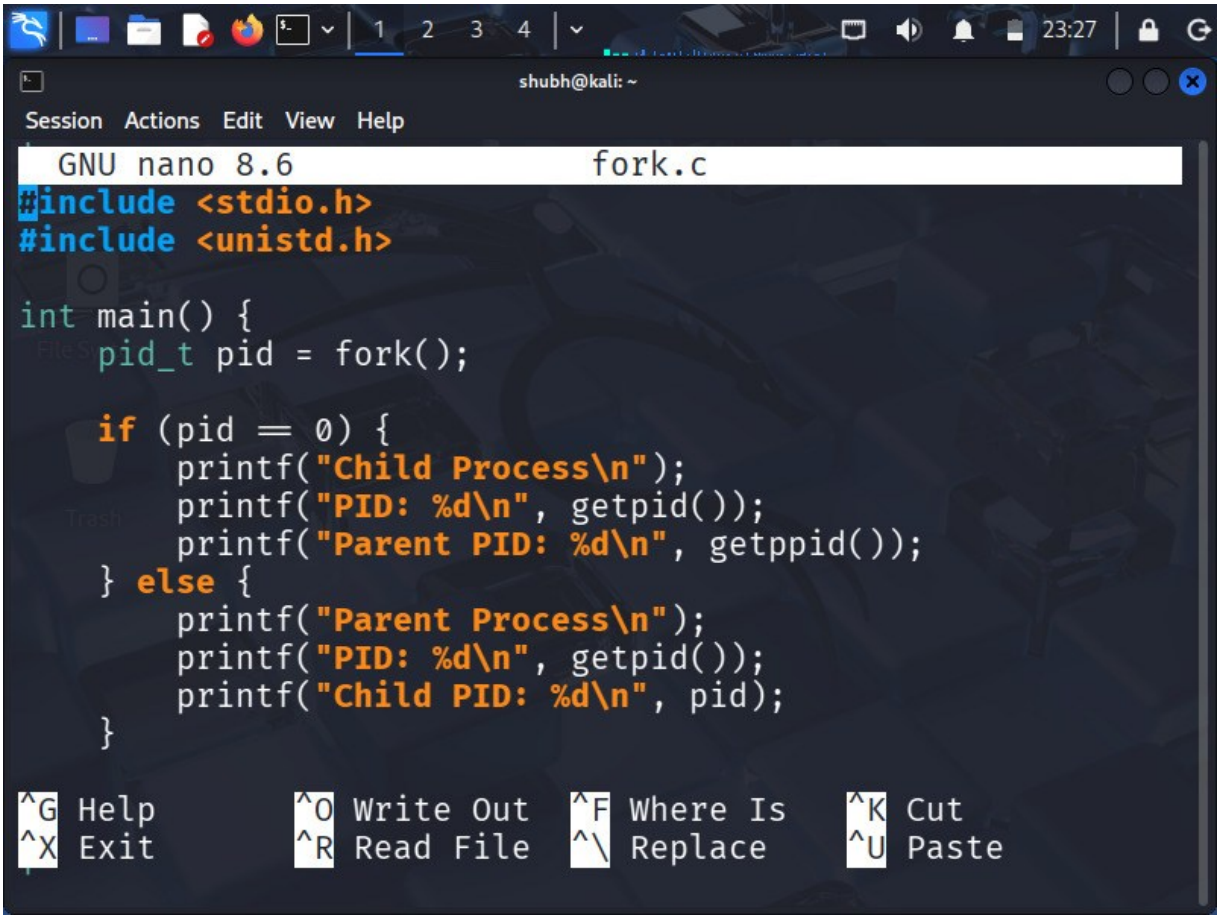
1.Child Process creation 2.Parent Process creation






1 2 3 4
23:19
shubh@kali: ~
Session Actions Edit View Help
(shubh㉿kali)-[~]
$ nano zombie.c
(shubh㉿kali)-[~]
$ gcc zombie.c -o zombie
(shubh㉿kali)-[~]
$ ./zombie
Parent Process ID: 21580
Child Process ID: 21581
(shubh㉿kali)-[~]
$ ./orphan
Parent Process ID: 22125
(shubh㉿kali)-[~]
$ Child Process ID: 22126
Parent Process ID: 1


1 2 3 4
23:28
shubh@kali: ~
Session Actions Edit View Help
nano fork.c
(shubh㉿kali)-[~]
$ nano fork.c
(shubh㉿kali)-[~]
$ gcc fork.c -o fork
(shubh㉿kali)-[~]
$ ./fork
Parent Process
PID: 26532
Child PID: 26533
Child Process
PID: 26533
Parent PID: 1
(shubh㉿kali)-[~]
$


1 2 3 4
23:27
shubh@kali: ~
Session Actions Edit View Help
GNU nano 8.6 fork.c
#include <stdio.h>
#include <unistd.h>
int main() {
pid_t pid = fork();
if (pid == 0) {
printf("Child Process\n");
printf("PID: %d\n", getpid());
printf("Parent PID: %d\n", getppid());
} else {
printf("Parent Process\n");
printf("PID: %d\n", getpid());
printf("Child PID: %d\n", pid);
}
^G Help ^O Write Out ^F Where Is ^K Cut
^X Exit ^R Read File ^\ Replace ^U Paste